

Supplement S1: Code Release for fitting OZAB and OZABE to Empirical Datasets in “Cohesive Framework for Modeling Plant Cover Class Data”

Written by Kathryn M. Irvine, US Geological Survey, kirvine@usgs.gov

June 5, 2019

In order to open this file and manipulate its contents, user needs an understanding of reproducible documents in RStudio (<https://www.rstudio.com/>). See guidance on knitr by RStudio at <https://yihui.name/knitr/>. The code to create the figures for the model posterior intervals is not shown in the .pdf, but is available in the .Rnw version of the file. This supplement contains the Stan code used to model the empirical examples using extensions to the ordinal zero-augmented beta model (OZAB) and with errors (OZABE). The posterior quantiles for each model are provided in Supplement S2.

The following files (datasets and model code) are required to run the code (easist to locate in the same directory as the .Rnw file):

- Multiple Observer Sagebrush Cover Class Data in Big Horn Canyon National Recreation Area , Montana-Wyoming 2015.csv: data file for BICA sagebrush example
- Double Observer Cover Class Data in Clarno unit John Day Fossil Beds, Oregon 2015.csv: data file for Clarno example with cover classes recorded on BareGrdCover, Bromus.tectorum, Elymus.caput.medusae, Poa.secunda, and Pseudoroegneria.spicata.
- ozabe.jags: JAGS version of the single species, simple random sampling design OZABE model
- ozab_multispp.stan: Stan implementation of the OZAB joint species model applied to Clarno dataset
- ozabe_multispp.stan: Stan implementation of the OZABE joint species model applied to Clarno dataset

- ozab_twostagedesign.stan: Stan implementation of the OZAB model to a two-stage cluster design applied to BICA sagebrush dataset
- ozabe_twostagedesign.stan: Stan implementation of the OZABE model to a two-stage cluster design applied to BICA sagebrush dataset

Verify the working directory is pointing to the correct location on your computer. The models take time to run on the empirical datasets (not seconds, but longest runtime was ≈ 17 minutes). Also, to recreate the .pdf flip all the chunks currently set to “eval = F” back to “T.” If this document does not compile *as is*, I suggest running each code chunk separately in Rstudio and uncommenting model summaries for a better look at the results.

```
## Code run and created on R version 3.5.1 (2018-07-02) -- "Feather Spray"

##Load packages
library(tidyr)
library(cowplot)
library(dplyr)
library(rstan)

## need a separate working JAGS instalation.

require(rjags) ; load.module('glm') ; load.module('dic')

##Set rstan to work on multiple cores
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

##set working session to the directory where the files were saved locally
setwd("D:/Current Papers/OZABE (in review)/Revision MEE 2019 May/S1 Supplement Info")
```

1 Example from Bighorn Canyon National Monument (BICA) with Heirarchical Sampling Design

```
##Analysis of BICA multi-observer ordinal cover data  
##includes both single and double observer plots.  
  
#these models assume a common detection model, vary mu, phi, and psi for each frame  
model_new3 <- stan_model('ozab_twostagedesign.stan')  
model_new4 <- stan_model('ozabe_twostagedesign.stan')  
  
#load data from local directory where downloaded from repository  
dat.subBICA.15<-  
  read.csv("Multiple Observer Sagebrush Cover Class Data in  
           Big Horn Canyon National Recreation Area , Montana-Wyoming 2015.csv",  
           header=TRUE)
```

1.1 Fitting OZAB to BICA dataset

```
#####  
##Prepare Artemisia.tridentata data for analysis using OZAB and OZABE models  
#####  
  
#indicator for detection/non-detection  
x <- ifelse(dat.subBICA.15$Artemisia.tridentata > 0, 1, 0)  
  
#number of unique plots  
N1 <- length(unique(dat.subBICA.15$uni.id))
```

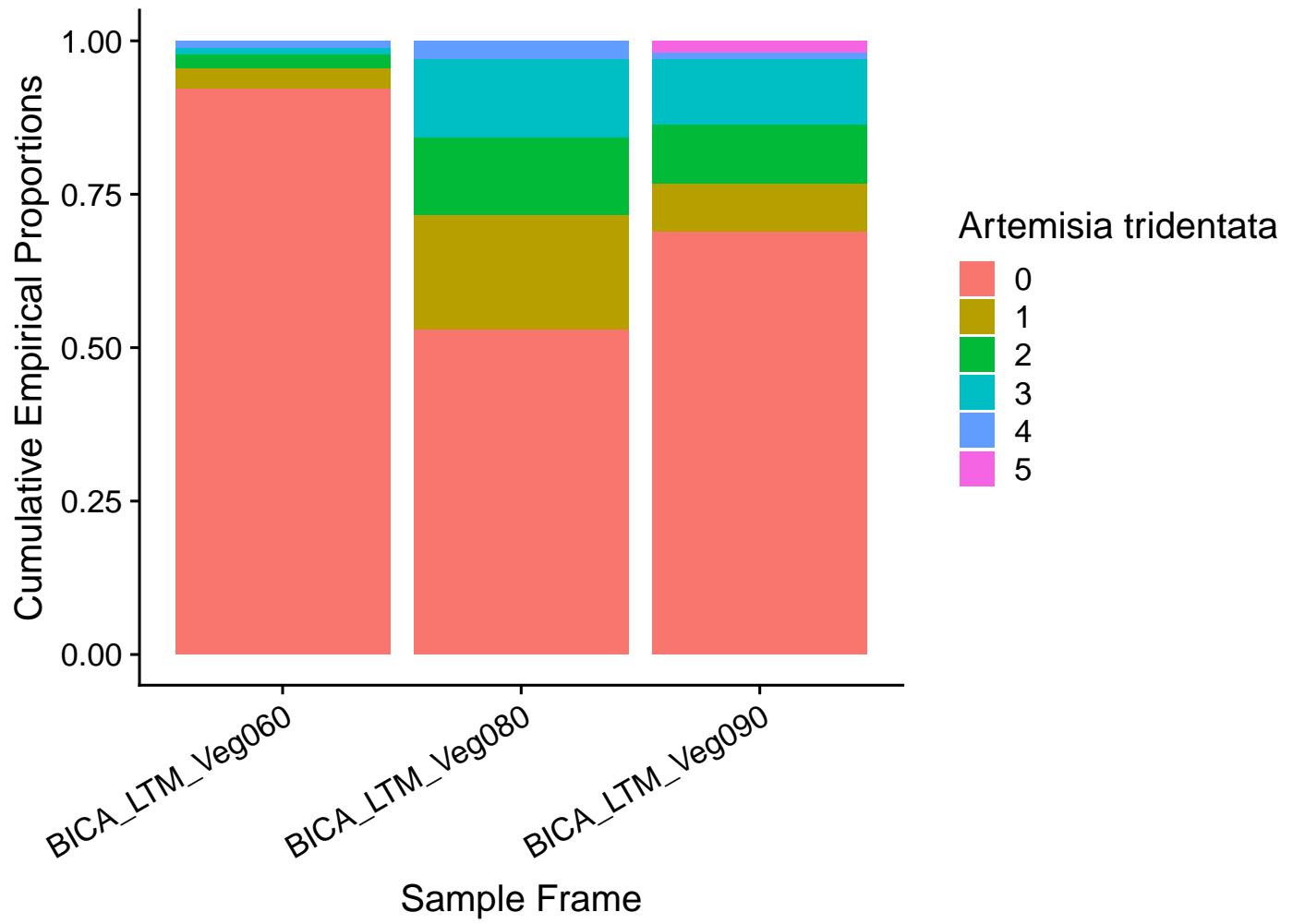


Figure S1.1: Stacked Barplot for BICA Sagebrush Example

```

#number of visits to each plot

visits1 <- as.vector(tapply(dat.subBICA.15$uni.id, dat.subBICA.15$uni.id, length))

#pulling off first element that starts the detection history to each plot

pos1 <- cumsum(c(1, visits1[-N1]))


#total number of all observations

n_obs1 <- sum(visits1)


#group id; here sample frame

lot.id <- as.vector(tapply(dat.subBICA.15$allot, dat.subBICA.15$uni.id, mean))


# number of groups

J <- length(unique(lot.id))


#number of plots with at least one detection

N2 <- sum(tapply(x, dat.subBICA.15$uni.id, max))



#number of visits with a detection per plot

visits2 <- as.vector(tapply(x[which(x > 0)],

                           dat.subBICA.15$uni.id[which(x > 0)],

                           length))



#indexing for the non-zero cover measurements

pos2 <- cumsum(c(1, visits2[-N2]))


#total number of non-zero observations

n_obs2 <- sum(visits2)


N1_ind <- as.vector(which(tapply(x, dat.subBICA.15$uni.id, max) > 0))

```

```

K <- 6 #number of categories

cuts <- c(0.05, 0.25, 0.5, 0.75, 0.95) #cutpoints on % cover scale

#ordinal values for observations with a detection
y_int <- dat.subBICA.15$Artemisia.tridentata[which(x > 0)]

##ignoring errors: OZAB model
#just first visit to each plot
z <- as.vector(tapply(x, dat.subBICA.15$uni.id, first))

#number of plots with a detection based on only first visit
N2_new <- sum(z)

#corresponding ordinal value for first visit to each plot
y_int2 <- as.vector(tapply(dat.subBICA.15$Artemisia.tridentata,
                           dat.subBICA.15$uni.id,
                           first))

#group level id for only plots with non-zeros
lot.id2 <- lot.id[which(y_int2 > 0)]

#removing the zeros from the ordinal variable
y_int2 <- y_int2[which(y_int2 > 0)]

#creating data input file for Stan model
sage.data2 <- list("N1" = N1,
                   "z" = z,
                   "N2" = N2_new,
                   "K" = K,
                   "cuts" = cuts,

```

```

"y_int" = y_int2,
"J" = J,
"grp_id" = lot.id,
"grp_id2" = lot.id2)

#fitting Stan OZAB model "ozab_twostagedesign"
sage.fit3 <- sampling(model_new3, sage.data2,control=list(adapt_delta=0.9))

#Returns posterior distribution summaries for the parameters specified in pars
# remove (#) to run code

#summary(sage.fit3, pars=c("mu", "phi", "psi"))$summary

#When I ran the model only 4 divergent transitions, ideally 0, but 4 is not too bad...
# A useful read about divergent transitions from Stan is located
#here https://mc-stan.org/users/documentation/case-studies/divergences\_and\_bias.html

# Warning messages:
# 1: There were 4 divergent transitions after warmup. Increasing adapt_delta above 0.9
#may help. See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
# 2: Examine the pairs() plot to diagnose sampling problems

```

1.2 Fitting OZABE to BICA dataset

```

#setting up data files for Stan model (ozabe_twostagedesign)
sage.data1 <- list("N1" = N1,
                   "visits1" = visits1,
                   "pos1" = pos1,
                   "n_obs1" = n_obs1,
                   "J" = J,

```

```

"grp_id" = lot.id,
"x" = x,
"N2" = N2,
"visits2" = visits2,
"pos2" = pos2,
"n_obs2" = n_obs2,
"K" = K,
"cuts" = cuts,
"y_int" = y_int)

#Fitting model ozabe_twostagedesign in Stan

sage.fit4 <- sampling(model_new4, sage.data1, control=list(adapt_delta=0.9))

#this model was the least stable & had sometimes too many divergent transitions,
#likely due to the sparsity of observation errors (hard to estimate sigma)
#and sparse cover classes (see Figure S2.1) for unit 1
# (hard to estimate mu[1], phi[1])

# Also, estimating phi (dispersion of the beta) could be improved,
# but the traceplots look ok and Rhat was <1.1

# #Warning messages:

# 1: There were 6 divergent transitions after warmup. Increasing adapt_delta above 0.9
#may help. See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
# 2: There were 4 chains where the estimated Bayesian Fraction of Missing Information
# was low. See
# http://mc-stan.org/misc/warnings.html#bfmi-low
# 3: Examine the pairs() plot to diagnose sampling problems

# the following code is commented out for compiling the document for easier reading

```

```

#summary(sage.fit4, pars=c("mu", "phi", "psi", "gamma0", "gamma1", "sigma"))#summary

#traceplots for mcmc chains

#stan_trace(sage.fit4, pars=c("gamma0", "gamma1", "sigma"))

# stan_trace(sage.fit4, pars=c("mu", "phi"))

# stan_trace(sage.fit4, pars=c("psi"))

#pairs plots

# pairs(sage.fit4,pars=c("mu", "psi", "phi"))

# pairs(sage.fit4,pars=c("gamma0", "gamma1", "sigma"))

```

2 Example for joint species modeling at Clarno unit of John Day Fossil Beds

```

#load Stan models from local folder

model_new3 <- stan_model('ozab_multispp.stan')

model_new5 <- stan_model('ozabe_multispp.stan')

#import data from csv file in local folder

dat.Clarno.spp1<-read.csv("Double Observer Cover Class Data in
                           Clarno unit John Day Fossil Beds, Oregon 2015.csv")

#need to put data into long format instead of short to allow for RE for each species...

dat.Clarno.spp<-

  dat.Clarno.spp1 %>% gather(species, "CoverClass", 5:9)

```

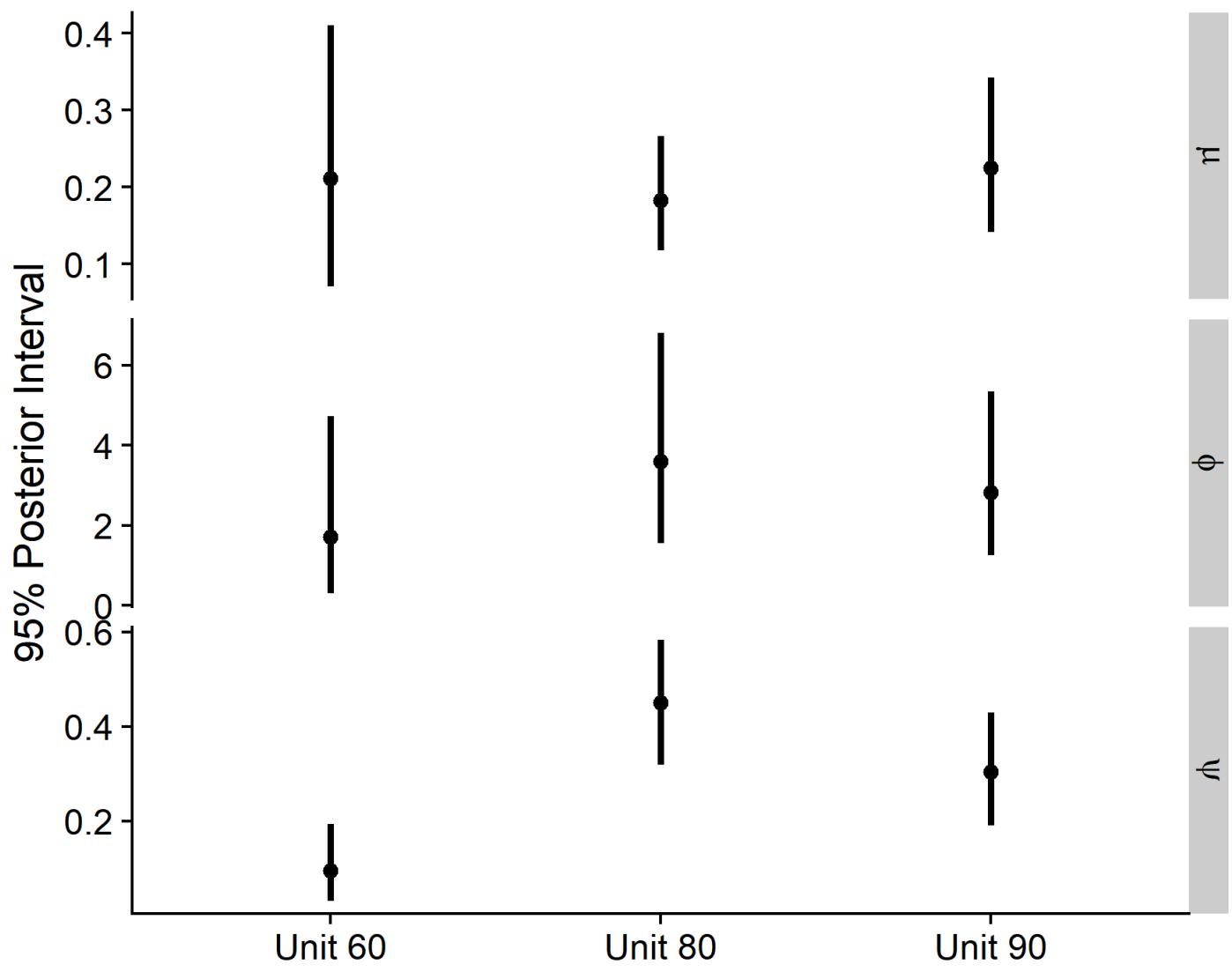


Figure S1.2: OZAB posterior intervals for parameters from hierarchical sampling design model fit to BICA dataset

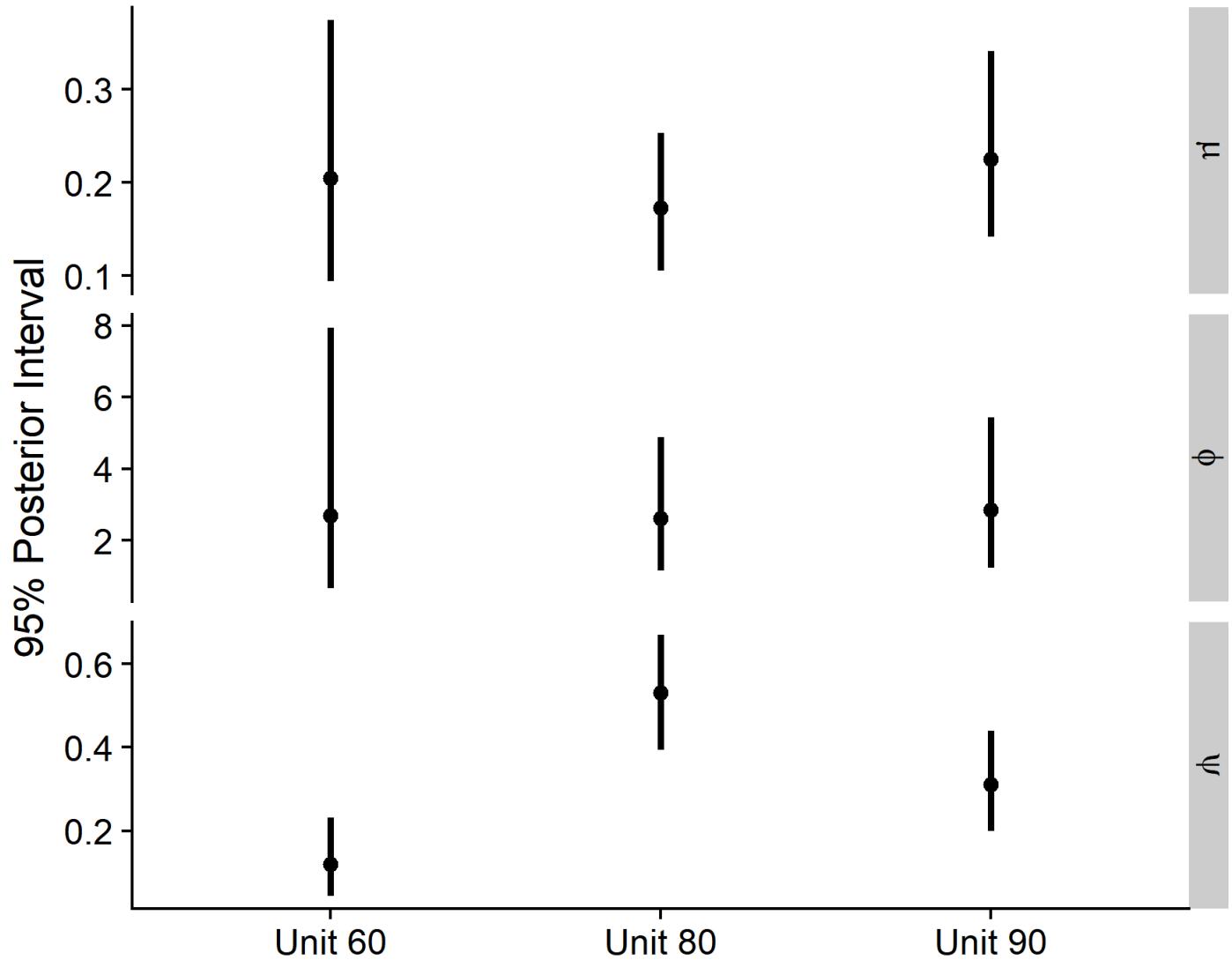


Figure S1.3: OZABE posterior intervals for parameters from two-stage design model fit to BICA dataset

```

#add detection indicator variable

dat.Clarno.spp$detect<- ifelse(dat.Clarno.spp$CoverClass>0,1,0)

#need to switch RE for allotment to species instead

#group-level, here species

dat.Clarno.spp$allot <- as.numeric(as.factor(dat.Clarno.spp$species))

# 220 unique plots with 160 single observer and 60 with double observer

# should have 280 rows in dat.Clarno.spp1

#should have 280 in the diagonal entries in the next call

#table(dat.Clarno.spp$species,dat.Clarno.spp$allot)

#filter to only first observation to each plot for fitting OZAB

dat.OZAB.Clarnospp<-dat.Clarno.spp %>%
  group_by(Loc_Name,species) %>%
  filter(row_number() == 1)

```

2.1 OZAB joint species model

```

#####
## Fitting multi-species OZAB model
#####

# group id for detection/non-det

grp.id <- dat.OZAB.Clarnospp$allot

# number of groups

```

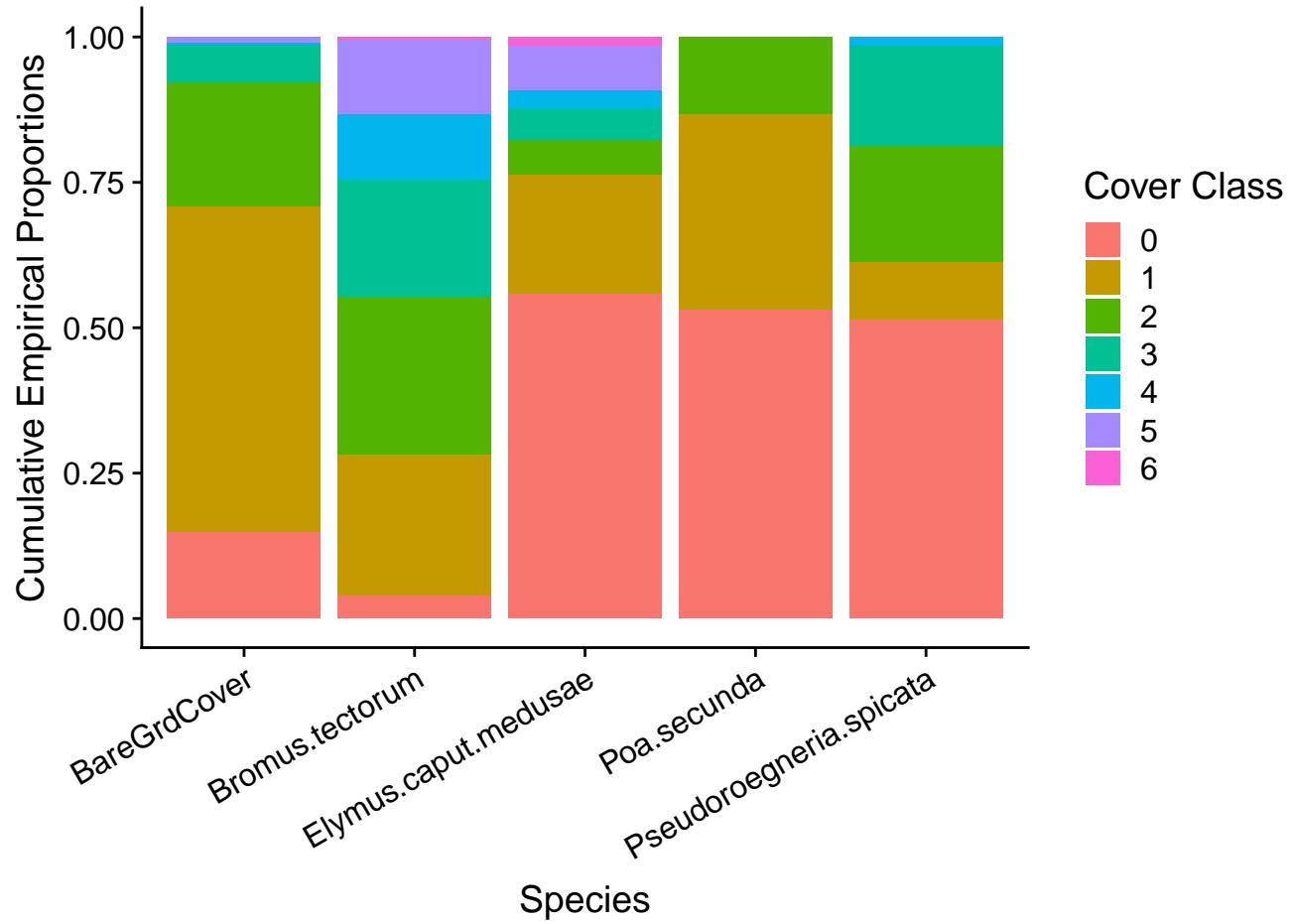


Figure S1.4: Cumulative empirical proportions by species surveyed at Clarno

```

J <- 5

N1<- length(dat.OZAB.Clarnospp$detect)

#just first visit to each plot for detection/non-detection

z <-dat.OZAB.Clarnospp$detect

#number of plots with a detection based on only first visit

N2_new <- sum(z)

#corresponding ordinal value for first visit to each plot

y_int2 <-dat.OZAB.Clarnospp$CoverClass

#group level id for only plots with non-zeros

grp.id2 <- dat.OZAB.Clarnospp$allot[which(y_int2 > 0)]

#removing the zeros from the ordinal variable

y_int2 <- y_int2[which(y_int2 > 0)]


#number of categories

K <- 6

#cutpoints on % cover scale

cuts <- c(0.05, 0.25, 0.5, 0.75, 0.95)

# Stan likes the data to be in the correct order...

claro.spp.data2 <- list("N1" = N1,
                         "z" = z,
                         "N2" = N2_new,
                         "K" = K,
                         "cuts" = cuts,
                         "y_int" = y_int2,
                         "J" = J,
                         "grp_id" = grp.id,
                         "grp_id2" = grp.id2)

```

```

#fitting the joint species OZAB model to Clarno dataset in Stan
clarno.spp.fit <- sampling(model_new3, clarno.spp.data2)

# Warning messages:
# 1: There were 4 divergent transitions after warmup. Increasing adapt_delta above 0.8
#may help. See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
# 2: Examine the pairs() plot to diagnose sampling problems
#
#
#summary(clarno.spp.fit, pars=c("mu", "phi", "psi"))

#pairs(clarno.spp.fit,pars=c("mu","phi","psi"))
#stan_trace(clarno.spp.fit,pars=c("mu","phi","psi"))

```

2.2 OZABE joint species model

```

#####
##Prepare data for a multi-species OZABE for CLARNO
#####

#create a unique id across plot*species combinations to assist with indexing
#should be ordered correctly, a critical step for the Stan model to work!!!
# suggest double-checking--
dat.Clarno.spp$uni.plotbySpp<-as.factor(paste(dat.Clarno.spp$species,
                                                 dat.Clarno.spp$Loc_Name,sep = "."))
#number of unique plots for each species

```

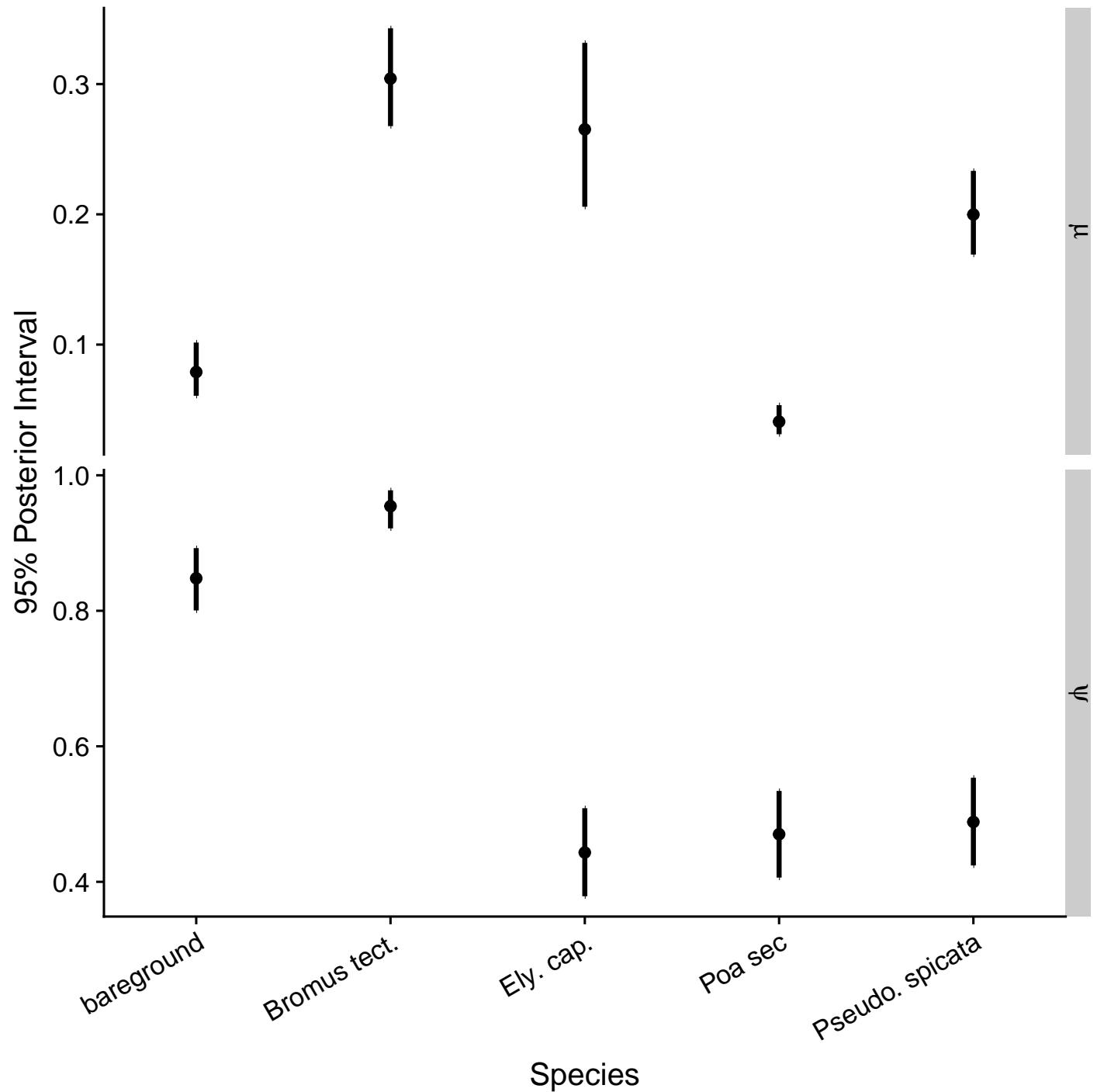


Figure S1.5: OZAB posterior intervals for parameters from joint species model fit to Clarno dataset

```

N1 <- length(unique(dat.Clarno.spp$uni.plotbySpp))

#vector with elements number of visits to each unique plot ordered by species
visits1 <- as.vector(table(dat.Clarno.spp$uni.plotbySpp))

#pulling off first element that starts the detection history to each plot
# concatenated by species
pos1 <- cumsum(c(1, visits1[-N1]))


#total number of all observations [rows in the long format dataset]
n_obs1 <- sum(visits1)


#indicator for detection/non-detection
x <- dat.Clarno.spp$detect


#group id: here identifies the data string for each species in longformat
grp.id <- as.vector(tapply(dat.Clarno.spp$allot, dat.Clarno.spp$uni.plotbySpp, mean))

# number of groups (5 species)
J <- length(unique(grp.id))

#number of plots with at least one detection: 715
N2 <- length(unique(dat.Clarno.spp$uni.plotbySpp[dat.Clarno.spp$detect>0]))


#vector with elements the number of visits with a detection per plot and species;
#length of vector same as N2
visits2 <- na.omit(as.vector(
  tapply(dat.Clarno.spp$uni.plotbySpp[which(dat.Clarno.spp$detect>0)], 
  dat.Clarno.spp$uni.plotbySpp[which(dat.Clarno.spp$detect>0)],length)))

```

```

#indexing for the non-zero cover measurements

pos2 <- cumsum(c(1, visits2[-N2]))


#total number of non-zero observations

n_obs2 <- sum(visits2)

N1_ind <- as.vector(c(1:N1)[tapply(dat.Clarno.spp$detect,
                                     dat.Clarno.spp$uni.plotbySpp, sum) > 0])

#number of categories

K <- 6


#cutpoints on % cover scale

cuts <- c(0.05, 0.25, 0.5, 0.75, 0.95)


#ordinal values for observations with a detection

y_int <- dat.Clarno.spp$CoverClass[dat.Clarno.spp$detect>0]

#creating dataset for Stan model

spp.data1 <- list("N1" = N1,
                  "visits1" = visits1,
                  "pos1" = pos1,
                  "n_obs1" = n_obs1,
                  "J" = J,
                  "grp_id" = grp.id,
                  "x" = x,
                  "N2" = N2,
                  "visits2" = visits2,
                  "pos2" = pos2,

```

```

  "n_obs2" = n_obs2,
  "K" = K,
  "cuts" = cuts,
  "y_int" = y_int)

# joint species OZABE model

# varying mu_s, psi_s [both non-centered parameterization],
# phi_s; gamma0_s both have independent priors across species s;
# fixing relationship between detection and latent abundance gamma1;
# common measurement error sigma among species

#THIS MODEL TAKES THE LONGEST...go make some tea [timed at 17min]
clarino.spp.fit4 <- sampling(model_new5, spp.data1,control=list(adapt_delta=0.85))

# summary(clarno.spp.fit4,
#         pars=c("mu", "phi", "psi", "gamma0", "gamma1", "sigma"))
# stan_trace(clarno.spp.fit4,pars="phi")
# stan_trace(clarno.spp.fit4,pars=c("gamma0", "gamma1", "sigma"))

# The following warning messages are likely due to estimating phi[1], bareground,
# phi[3]= Elycap, phi[4]=Poa sec, phi[5]= pseudo spic, see traceplots
# R-hats and other traceplots look ok; this could be a good case for changing
# the priors on phi to improve mixing, but good enough as an example

# Warning messages:
# 1: There were 3 divergent transitions after warmup. Increasing adapt_delta above 0.85
# may help. See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

```

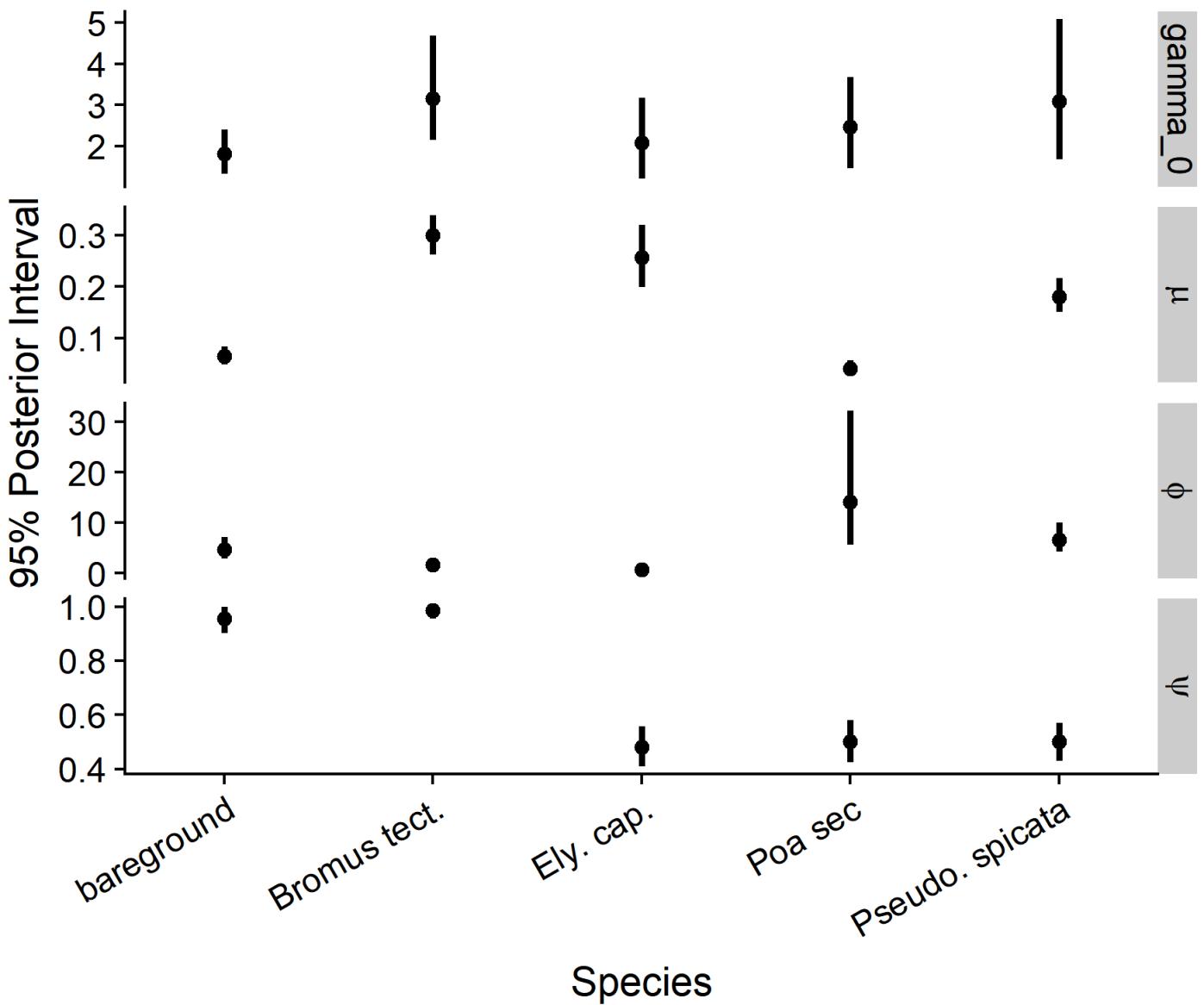


Figure S1.6: OZABE posterior intervals for parameters from joint species model fit to Clarno dataset

```
# 2: There were 4 chains where the estimated Bayesian Fraction of Missing Information
# was low. See http://mc-stan.org/misc/warnings.html#bfmi-low
# 3: Examine the pairs() plot to diagnose sampling problems
```

3 JAGS OZABE Code Implementation with Simulated Data

The following code chunk simulates data under OZABE as shown in Fig. 1 in the main paper and then contains code to fit the JAGS version of OZABE.

```

#this fits the OZABE model to simulated data with JAGS

##simulate some data

mu.true <- 0.25
phi.true <- 3
psi.true <- 0.5
sigma.true <- 0.5
gamma0.true <- -1
gamma1.true <- 4

##process model

y.true <- rbinom(400, 1, psi.true) * rbeta(400, mu.true*phi.true, (1-mu.true)*phi.true)

##observation model

x.mat <- matrix(NA, ncol=6, nrow=400)
for(i in 1:400){
  x.mat[i, ] <- (y.true[i] > 0) *
    rbinom(6, 1, plogis(gamma0.true + gamma1.true*y.true[i]))
}

u.mat <- matrix(NA, ncol=6, nrow=400)
for(i in 1:400){
  for(j in 1:6){
    u.mat[i, j] <- ifelse(x.mat[i, j] > 0,
      plogis(rnorm(1, qlogis(y.true[i]), sigma.true)), 0)
  }
}

```

```

#functions needed to fit JAGS model

cumsum.no.zero <- function(x){
  temp <- cumsum(x)
  temp[which(temp==0)] <- 1
  return(temp)
}

#creates observed cover classes with measurement error and NA for no detections

u.mat.fun <- function(x.mat, u.mat){
  temp <- t(apply(u.mat, 1, sort, decreasing=TRUE))
  temp[which(temp!=0)] <- cut(temp[which(temp!=0)],
                                c(0, 0.10, 0.25, 0.50, 0.75, 1.00),
                                labels=FALSE)
  temp[which(temp==0)] <- NA
  return(temp[which(rowSums(x.mat) > 0), ])
}

#cutpoints for interval or cover classes

cuts <-c(0.05, 0.25, 0.50, 0.75, 0.95)

#formatting data for JAGS model

jags.dat <- list(x      = x.mat,
                  N1     = nrow(x.mat),
                  N2     = sum(apply(x.mat, 1, max)),
                  y.ind = ifelse(rowSums(x.mat) >0, 1, 0),
                  y.use = cumsum.no.zero(ifelse(rowSums(x.mat) > 0, 1, 0)),
                  V1     = rep(6, nrow(x.mat)),
                  V2     = apply(x.mat[which(rowSums(x.mat) > 0), ], 1, sum),
                  obs.c = u.mat.fun(x.mat, u.mat),

```

```

cuts  = c(0, cuts))

#initial values

midpoints <- c(0.025, 0.15, 0.375, 0.625, 0.85, 0.975)

y.init <- midpoints[apply(u.mat.fun(x.mat, u.mat),
                           1, max, na.rm=TRUE)]

u.temp <- u.mat.fun(x.mat, u.mat)

log.u.init <- matrix(NA, ncol=ncol(u.temp), nrow=nrow(u.temp))

for(i in 1:nrow(u.temp)){
  for(j in 1:ncol(u.temp)){
    log.u.init[i, j] <- ifelse(is.na(u.temp[i, j]) == TRUE, NA,
                                qlogis(midpoints[u.temp[i, j]]))
  }
}

inits4 <- function(){list(psi      = runif(1),
                           mu       = runif(1, 0.2, 0.8),
                           phi      = rgamma(1, 3, 1),
                           gamma0  = rnorm(1, 0, 4),
                           gamma1  = rnorm(1, 0, 4),
                           z        = apply(x.mat, 1, max),
                           sigma   = rgamma(1, 2, 1),
                           y       = y.init,
                           log.u  = log.u.init,
                           y.new  = rep(0.5, nrow(x.mat)))}

```

```
#fit the OZABE model

jags4 <- jags.model("ozabe.jags", jags.dat, inits4, n.chains=4, n.adapt=5000)
jags4.out <- coda.samples(jags4, c("mu", "phi", "psi", "gamma0",
                                    "gamma1", "sigma"), n.iter = 5000, thin=5)

gelman.diag(jags4.out)
summary(jags4.out)
```